

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SLEDOVÁNÍ PAPRSKU VE 2D PŘES JEDNODUCHÉ OPTICKÉ ČLENY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DÁVID DANCZI

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SLEDOVÁNÍ PAPRSKU VE 2D PŘES JEDNODUCHÉ OPTICKÉ ČLENY

BASIC OPTICAL ELEMENTS 2D RAYTRACING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DÁVID DANCZI

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAL SEEMAN

BRNO 2010

Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem a implementací programu, který simuluje metodu sledování paprsku přes jednoduché optické členy. Dále rozebírá problematiku základů počítačové grafiky a popis programového rozhraní Windows API.

Abstract

This bachelor's thesis describes analysis, design and implementation of an application, which simulates the method of ray tracing through simple optical elements. Further it examines the issue of computer graphics and description of the Windows API programming interface.

Klíčová slova

optika, zrcadlo, čočka, paprsek, světlo, odraz, lom, sledování paprsku, optické členy

Keywords

optics, mirror, lens, ray, light, reflection, refraction, ray tracing, optical elements

Citace

Dávid Danczi, Sledování paprsku ve 2D přes jednoduché optické členy, bakalářská práce, Brno, FIT VUT v Brně, 2010

Sledování paprsku ve 2D přes jednoduché optické členy

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Michala Seemana.

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Dávid Danczi
19.5.2010

Pod'akovanie

Chcel by som poďakovať môjmu školiťovi Ing. Michalovi Seemanovi, mojim rodičom, bratovi, Marekovi Hlinkovi, Petre Zádekovej, Tomášovi Škorňákovi, Michalovi Krupovi, Michalovi Černákovi, Romanovi Kantorovi a Tomášovi Kaňokovi.

© Dávid Danczi, 2010

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
2 Teoretický úvod.....	3
2.1 Optika.....	3
2.2 Svetlo.....	3
2.3 Odraz a lom svetla.....	4
2.4 Geometrická optika.....	5
2.4.1 Zobrazenie rovinným zrkadlom.....	6
2.4.2 Guľové zrkadlo.....	6
2.4.3 Zobrazovanie dutým zrkadlom.....	7
2.4.4 Zobrazovanie vypuklým zrkadlom.....	7
2.4.5 Šošovky ako zobrazovacie sústavy.....	8
3 Počítačová grafika.....	9
3.1 Základné prvky grafiky.....	9
3.2 Orezávanie.....	10
3.3 Ray-casting.....	10
3.4 Raytracing.....	10
3.5 Vzájomná poloha objektov a lúčov.....	12
3.5.1 Prienik dvoch priamok v 2D.....	12
3.5.2 Prienik priamky a kružnice v 2D.....	13
4 Win32 API.....	14
5 Analýza a návrh riešenia.....	15
6 Implementácia.....	16
7 Záver.....	22
7.1 Možnosti rozšírenia.....	22
Literatúra.....	23
Zoznam príloh.....	24

1 Úvod

Vývoj počítačovej grafiky sa v dnešnej dobe značne urýchlil. Čím ďalej, tým viac sa virtuálne zobrazenie na počítačoch blíži k reálnemu zobrazeniu skutočného sveta. Vizualizácia pomáha pri konštrukcii rôznych technických zariadení, v medicíne, pri simuláciách, zobrazuje virtuálny svet počítačových hier. Počítačová grafika uľahčuje ľuďom komunikáciu so strojmi pri každodennej práci v mnohých odvetviach.

Búrlivý rozvoj moderného sveta kladie stále vyššie nároky na človeka a na jeho zmyslové orgány, ktoré sprostredkujú spojenie jedinca s obklopujúcim materiálnym svetom. Pre svoju činnosť potrebuje centrum nervového systému – mozog, kvalitné informácie, ktoré umožnia poznávať okolitý svet v plnej šírke. Sprostredkovateľom týchto informácií sú zmyslové orgány človeka, prispôsobené na príjem rôznych podráždení a ich prenos do mozgu vo forme biosignálov. Človek kontroluje svoje pohyby a svoju činnosť predovšetkým orgánom zraku – očami a tento zmyslový orgán nadobúda u neho rozhodujúci význam pre praktický život. Správna funkcia orgánu zraku je základným predpokladom získania vnemov z vonkajšieho materiálneho sveta a tým aj správne reagovať na meniace sa podmienky okolia. Ak nastane chyba v zraku človeka, napr. krátkozrakosť alebo ďalekozrakosť, v dnešnej dobe sa dá tento problém vyriešiť pomocou základných optických prvkov. Sú to šošovky typu spojka, rozptylka. Tak isto môže nastať situácia, že človek je na križovatke a vedľajšia cesta má príliš veľký uhol, aby jedinec videl, či môže vo svojej ceste pokračovať, alebo musí počkať na okoloidúce autá. Na tieto účely slúžia rovné, duté alebo vypuklé zrkadlá. Úlohou týchto prvkov je zobrazenie obrazu reality pomocou odrazu alebo lomu svetelného lúča.

Touto prácou chcem poukázať na to, ako fungujú jednoduché optické prvky a jednoduchou simuláciou zobrazit' správanie sa svetelného lúča, ktorý narazí na ich povrch. Tento lúč bude pochádzať z umelo vytvoreného svetelného zdroja a jeho smer sa pri každom náraze na optické prvky zmení. Zmena smeru nastáva odrazom, ak je prvok zrkadlo, poprípade sa lúč lomí akonáhle narazí na šošovku.

Práca sa zaoberá predstavením programovacieho jazyka C, C++, WinAPI a metódu zobrazenia svetla, jeho odrazu a lomu na rozhraní objektov.

2 Teoretický úvod

2.1 Optika

Optika je náuka o videní. Skúma povahu svetla, zákonitosti svetelných javov, vzájomné pôsobenie látky a poľa. Svetlo je elektromagnetické vlnenie, ktoré charakterizujú vektor elektrickej intenzity E a vektor magnetickej indukcie B . Tieto vektory kmitajú v dvoch rovinách na seba kolmých a zároveň sú kolmé aj na smer súvisiaceho sa vlnenia na c .

2.2 Svetlo

Svetlo je elektromagnetické vlnenie so zložkou tzv. Elektromagnetického spektra. Je to súbor elektromagnetických vlnení s rozsahom vlnových dĺžok od 10^8m až do 10^{-16}m . Viditeľné svetlo (vyvoláva zrakový vnem) má rozsah od 780 do 380nm ($1\text{nm}=10^{-9}\text{m}$). Oko je najcitlivejšie na vlnovú dĺžku 555nm, čo vyjadruje žltozelené svetlo. Vznik viditeľného svetla súvisí s kvantovými procesmi, prebiehajúcimi v elektrónových obaloch atómov.

Teleso, ktoré emituje svetlo sa nazýva svetelný zdroj. Svetelné zdroje rozoznávame: prirodzené (hviezdy, slnko), umelé (žiarovka, žiarivka), chromatické, ktoré sú zložené z viacerých vlnových dĺžok a monochromatické, ktoré majú jednu vlnovú dĺžku.

Prostredie, v ktorom sa svetlo od zdroja šíri sa nazýva optické prostredie. Hmotné prostredie svetlo viac alebo menej absorbuje (pohlcuje). Podľa toho rozlišujeme prostredia:

1. Priehľadné prostredie, ktoré svetlo dobre prepúšťa, absorbuje ho minimálne a cez takéto prostredie predmety dobre vidíme (vzduch, voda).
2. Nepriehľadné prostredie, svetlo neprepúšťa, odráža ho, alebo pohlcuje. Cez takéto prostredie predmety nevidíme. (kovy, drevo, kameň)
3. Priesvitné prostredie, svetlo prepúšťa, ale rozptyľuje ho v rôznych smeroch (mliečne sklo).

Prostredie, ktoré má vo všetkých smeroch rovnaké vlastnosti sa nazýva rovnorodé, homogénne prostredie. Svetlo sa šíri priamočiaro. Ak uvažujeme v rovnorodom, homogénnom prostredí prítomnosť bodového zdroja, svetlo sa šíri vo forme guľových vlnoplôch.

Platí Huygensoov princíp. Vo veľkej vzdialenosti od zdroja, časť guľového vrtlíka môžeme považovať za rovinu a hovoríme o rovinnej vlnoploche. Šíriace sa svetlo v priestore znázorňujeme pomocou svetelných lúčov. Je to normálna (kolmica) k vlnoploche.

Svetlo sa šíri najväčšou rýchlosťou vo vákuu

$$c = 2,997\,924\,58 \cdot 10^8 \text{ m} \cdot \text{s}^{-1}$$

Rýchlosť svetla vo vákuu je hraničnou rýchlosťou (medzná rýchlosť) vo vesmíre. V hmotnom prostredí sa svetlo šíri pomalšie než vo vákuu. Ak je svetlo hmotným prostredím málo absorbované (prostredie svetlo dobre prepúšťa) a jeho rýchlosť v danom prostredí je o málo menšia od rýchlosti vo vákuu, takéto hmotné prostredie nazývame opticky redšie prostredie. Hmotné prostredie, v ktorom je svetlo viac pohlcované je opticky hustejšie. (menej prepúšťa svetlo, šíri sa menšou rýchlosťou v porovnaní s vákuom).

Stáva sa, že zobrazovaný predmet nemôžeme priamo pozorovať voľným okom. Lúče z daného predmetu musia najskôr odrazom alebo lomom zmeniť smer, kým dopadnú na sieťnicu oka. Sú to

predmety veľmi ďaleké, veľmi blízke, alebo predmety, ktoré je nemožné pozorovať priamo z oka, napr. vlastné ucho. V týchto prípadoch lúče nepochádzajú z bodu A, ale zo skutočného obrazu A', poprípade z neskutočného obrazu A". Skutočný obraz A' vznikne, ak lúče vplyvom optickej sústavy vytvoria zbiehavý zväzok, ktorý sa stretne v ich priesečníku. Neskutočný obraz vznikne, ak lúče vytvoria rozbiehavý zväzok. Naše oko je schopné tieto lúče spojiť do jedného bodu. Ohnisko je bod na optickej osi, v ktorom sa pretnú všetky rovnobežné lúče s osou, alebo bod, z ktorého lúče po prechode spojnou šošovkou vytvoria zväzok lúčov rovnobežný s optickou osou. Prostredia, ktoré zmenili smer prichádzajúceho lúča nazývame optické prostredia.

2.3 Odraz a lom svetla

Keď dopadá svetlo na povrch, dopadá na rozhranie dvoch prostredí. Časť svetla sa odráža od povrchu, časť sa v telese pohlcuje a pridá telesu energiu, časť svetla prechádza do druhého prostredia. Časť svetla, ktorá sa odrazila, zostáva v rovine dopadu a jeho vlnová dĺžka sa nezmenila. Tieto vlastnosti vyjadruje zákon odrazu. Zákon odrazu znie: Svetlo sa na rozhraní dvoch prostredí odráža pod takým istým uhlom, pod akým dopadlo. Uhol odrazu sa rovná uhlu dopadu. Odrazený lúč leží v tej istej rovine ako dopadnutý lúč.

Lom svetla nastáva vtedy, ak svetlo prechádza rozhraním z jedného prostredia do druhého. Lomom svetla nastáva zmena jeho rýchlosti, a aj zmena smeru šírenia svetla. Pretože optické prostredie viac alebo menej svetlo pohlcuje a svetlo sa šíri v rôznych optických prostrediach, rôznou rýchlosťou, zavádzame veličinu absolútny index lomu prostredia (látky).

Absolútny index lomu n je daný pomerom rýchlosti svetla vo vákuu a rýchlosti svetla v danom hmotnom prostredí. $n=c/v$, kde n je bezrozmerná veličina, udáva koľkokrát sa svetlo šíri v hmotnom prostredí menšou rýchlosťou než vo vákuu.

Lom ku kolmici nastáva, ak svetlo prechádza z opticky redšieho prostredia do opticky hustejšieho. Lom od kolmice nastáva ak svetlo prechádza z opticky hustejšieho prostredia do prostredia redšieho.

Pre lom svetla platí zákon lomu

1. Pomer sínusov uhla dopadu a uhla lomu sa rovná pomeru rýchlostí ktorými sa svetlo šíri v daných prostrediach.

$$\frac{\sin \alpha}{\sin \beta} = \frac{v_1}{v_2} \quad (1)$$

$$n_1 = \frac{c}{v_1} \quad n_2 = \frac{c}{v_2}$$

$$\frac{\sin \alpha}{\sin \beta} = \frac{n_2}{n_1}$$

Snellov zákon lomu: Pomer sínusov uhla dopadu a uhla lomu sa rovná obrátenému pomeru absolútnych indexov lomu daných prostredí (jednotlivých prostredí), ktorými sa svetlo šíri.

Pri lome od kolmice veľkosť uhla dopadu nemôže byť ľubovoľná. Aby nastal lom svetla najväčšia hodnota ktorú uhol dopadu môže nadobudnúť je tzv. medzná hodnota α_m

Ak svetlo dopadá pod uhlom väčším ako α_m , nastáva úplný odraz svetla.

Odvodenie zákona lomu: Ak svetlo dopadá na rozhranie pod medzným uhlom α_m a opticky redším prostredím (prostredie n_2) je vzduch, platí :

$$\frac{\sin \alpha}{\sin \beta} = \frac{n_2}{n_1}$$

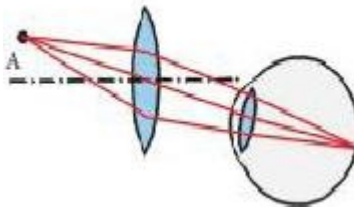
$$\alpha = \alpha_m, \beta = 90^\circ, \sin \beta = 1, n_2 = 1$$

$$\frac{\sin \alpha}{1} = \frac{1}{n_1}$$

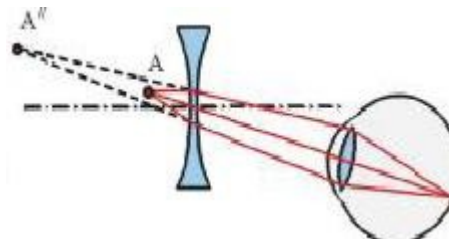
Na tomto princípe sú založené refraktometre, prístroje určené na meranie indexu lomu pevných a kvapalných látok, pri známej hodnote medzného uhla α_m . Úplný odraz sa využíva pri odrazových hranoloch ktoré sú súčasťou mnohých optických prístrojov.

2.4 Geometrická optika

Optická sústava je sústava optických prostredí a ich rozhraní, ktorá mení smer chodu lúčov. Postup, ktorým získavame obrazy predmetov sa nazýva optické zobrazovanie. Pri zobrazovaní môžeme získať reálny, alebo zdanlivý (virtuálny) obraz predmetu.



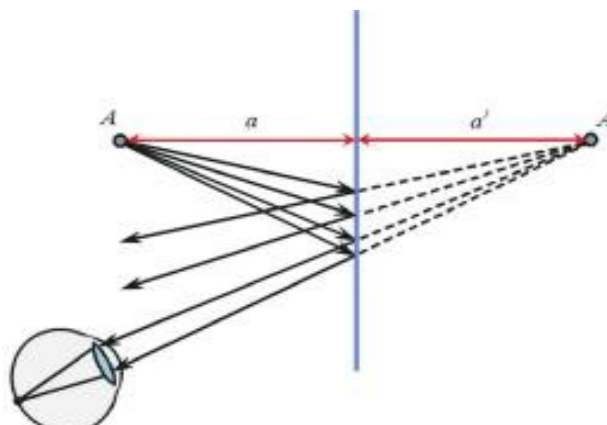
Obrázok 1: Skutočný obraz



Obrázok 2: Virtuálny obraz

2.4.1 Zobrazenie rovinným zrkadlom

Lúče dopadajúce z bodu A na rovinné zrkadlo sa odrážajú (platí zákon odrazu). Z každého bodu predmetu vychádza rozbiehavý zväzok lúčov. Odrazené lúče sú tak isto rozbiehavé a pozorovateľovi sa zdá, že lúče vychádzajú spoza zrkadla z neskutočného bodu A', lúče ani žiarivá energia ním neprechádzajú. Obraz vytvorený rovinným zrkadlom je neskutočný, priamy, rovnako veľký ako zobrazovaný predmet a je súmerne zlúčený s predmetom podľa roviny zrkadla. Rovinné zrkadlá sú používané v bežnej praxi v každodennom živote. Majú zrkadliacu plochu (cínový amalgám) nanesenú na zadnej strane zrkadla.



Obrázok 3: Zobrazenie rovinným zrkadlom

2.4.2 Guľové zrkadlo

Guľové zrkadlo má charakteristické tieto vlastnosti: Stred guľovej plochy je stred krivosti zrkadla, označuje sa písmenom C. Polomer guľovej plochy sa nazýva polomer krivosti. Stred zrkadliacej plochy nazývame vrchol V a priamka ktorá prechádza stredom krivosti a vrcholom je optická os zrkadla. V strede vzdialenosti vrchola a stredom krivosti sa nachádza ohnisko zrkadla F. Ohnisko je obraz osového predmetu nachádzajúceho sa vo veľmi veľkej vzdialenosti, teoreticky v nekonečne pred zrkadlom. Duté zrkadlo má skutočné ohnisko, vypuklé zrkadlo má ohnisko neskutočné. Vzdialenosť ohniska od vrcholu zrkadla sa nazýva ohnisková vzdialenosť.

Polohu a vlastnosti obrazu zistíme pomocou

1. zobrazovacej rovnice guľového zrkadla,
2. rovnice priečného zväčšenia,

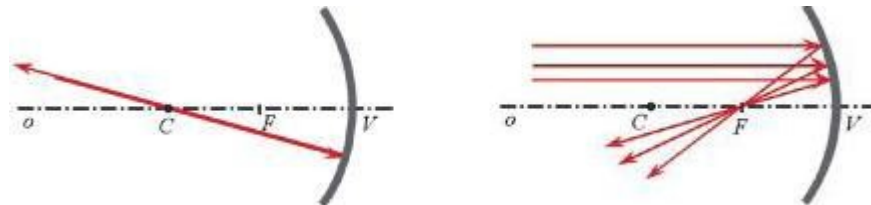
Pri zobrazovaní odrazom na guľovej ploche využívame paraxiálne lúče. Sú to lúče, ktoré sa šíria v blízkosti optickej osi a sú s ňou rovnobežné. Tieto lúče spadajú do tzv. Paraxiálneho priestoru zrkadla. Pri sledovaní lúča pri odraze guľovým zrkadlom môžu nastať tieto situácie:

1. Lúč prechádzajúci stredom krivosti zrkadla dopadá na zrkadlo kolmo a odráža sa smerom opačným.

2. Lúč rovnobežný s optickou osou (paraxiálny lúč) po odraze zrkadlom smeruje do ohniska zrkadla.
3. Lúč prechádzajúci ohniskom má po odraze smer rovnobežný s optickou osou.

Poznáme dva typy guľových zrkadiel, duté a vypuklé.

2.4.3 Zobrazovanie dutým zrkadlom



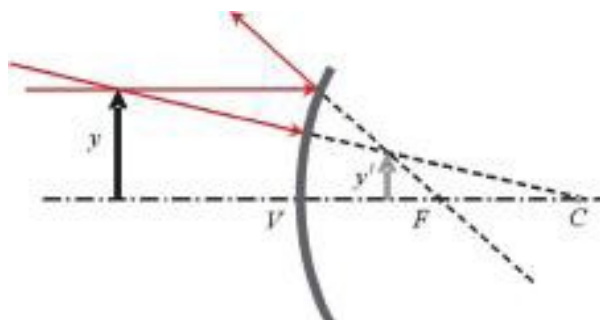
Obrázok 4: Odraz lúča dutým zrkadlom

Pri zobrazovaní predmetu guľovým zrkadlom platia tieto pravidlá:

1. Ak vzdialenosť predmetu od zrkadla je väčšia ako dvojnásobok ohniskovej vzdialenosti zrkadla, obraz je skutočný, prevrátený a zmenšený (duté zrkadlo).
2. Ak je rovné dvojnásobku ohniskovej vzdialenosti obraz je skutočný, prevrátený, rovnako veľký (duté zrkadlo).
3. Ak sa predmet nachádza pred zrkadlom vo vzdialenosti väčšej ako ohnisková a menšej ako dvojnásobok ohniskovej vzdialenosti obraz je skutočný, prevrátený a zväčšený.
4. Ak sa predmet nachádza pred zrkadlom vo vzdialenosti rovnajúcej sa ohniskovej vzdialenosti čiže $a=f$ odraz vzniká v nekonečne.
5. Ak je $a < f$, čiže predmet sa nachádza pred zrkadlom vo vzdialenosti menšej ako ohnisková vzdialenosť, obraz je neskutočný priamy a zväčšený. Je to jediný prípad keď dutým zrkadlom vytvoríme neskutočný obraz predmetu.

2.4.4 Zobrazovanie vypuklým zrkadlom

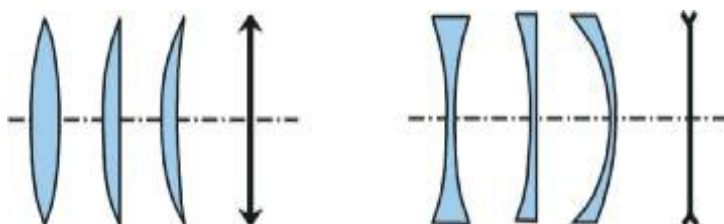
Pri zobrazovaní vypuklým zrkadlom získavame obraz predmetov, ktoré sú vždy neskutočné, priame a zmenšené.



Obrázok 5: Zobrazovanie vypuklým zrkadlom

2.4.5 Šošovky ako zobrazovacie sústavy

Šošovky sú opticky priehľadné prostredia ohraničené väčšinou guľovými plochami. Stredy týchto plôch nazývame stredy krivosti a označujú sa S_1 a S_2 . Priamka ktorá spája stredy krivosti je optická os šošovky. V praxi sa najčastejšie používajú duté šošovky – rozptylky a vypuklé – spojky.



Obrázok 6: Modely spojných a rozptylných šošoviek

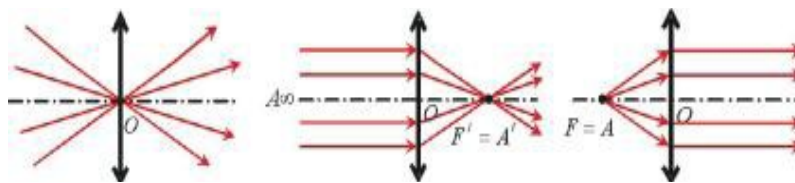
Poznámka: pri riešení úloh uvažujeme o tenkých šošovkách. Tenké šošovky sú také, ktorých hrúbku možno zanedbať a teda optický stred šošovky možno stotožniť s vrcholmi oboch guľových plôch. $V_1=0=V_2$.

Ohniská spojky sú skutočné, ohnisko je obraz bodového osového predmetu nachádzajúceho sa vo veľmi veľkej vzdialenosti od šošovky. Rozptylka má neskutočné ohniská, dajú sa zostrojiť len geometricky. Žiarivá energia týmito bodmi neprechádza. Vzdialenosť optického bodu šošovky a predmetového ohniska šošovky je predmetová ohnisková vzdialenosť F . $-|OF| = f$, a tiež $|OF'| = f$.

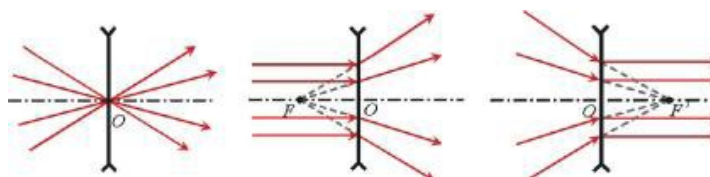
Vzdialenosť optického stredy a obrysového ohniska šošovky je obrazová ohnisková vzdialenosť. Ak prostredie pred šošovkou a prostredie za šošovkou je rovnaké, predmetové ohnisko a obrazové ohnisko sa rovnajú, preto používame len pojem ohnisková vzdialenosť šošovky.

Pravidlá pre prechod lúča šošovkami:

1. Lúč, ktorý prechádza cez optický stred šošovky nemení svoj smer, šíri sa priamočiario.
2. Lúč ktorý dopadá na šošovku cez ohnisko sa po prechode šošovkou, po lome šíri rovnobežne s optickou osou. Tento lúč sa využíva len pri zobrazení spojkou.
3. Lúč, prechádzajúci ohniskom šošovky sa po dopade na šošovku bude šíriť rovnobežne s optickou osou.



Obrázok 7: Prechod lúča spojnou šošovkou



Obrázok 8: Prechod lúča rozptylnou šošovkou

3 Počítačová grafika

Prevod základných grafických prvkov do postupnosti obrazových bodov nazývame rasterizácia.

3.1 Základné prvky grafiky

Základným prvkom rasterizácie je bod, z ktorého sú poskladané všetky objekty. Pri rasterizácii vzniká chyba zobrazovania, ktorá je spôsobená veľkosťou pixelu, ktorý je najmenšou jednotkou v grafike, musí mať celočíselnú hodnotu a preto nastávajú zaokrúhľovacie chyby.

Úsečka je základný grafický prvok, pomocou ktorého sa vykresľujú zložitejšie objekty. Kým úsečka je v analytickej matematike nekonečná množina bodov, v grafike je omedzená na určitý počet bodov, ktoré je možné zobrazit' na počítači (monitore). Úsečka sa najčastejšie zadáva pomocou začiatkových a koncových bodov $[x_1, y_1][x_2, y_2]$, prípadne pomocou bodu a vektorom rozdielu súradníc $[x_1, y_1][x_2 - x_1, y_2 - y_1]$. Smernicová rovnica úsečky je vyjadrená vzorcom

$$(y - y_0) = k * (x - x_0) + q$$

kde smernica k sa dá vyjadriť ako

$$k = \frac{(y - y_0)}{(x - x_0)}$$

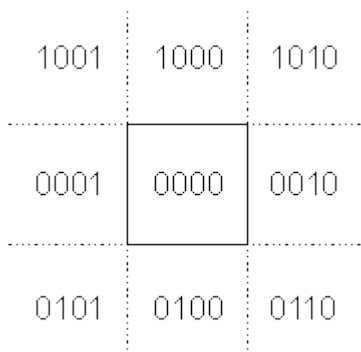
Pomocou smernice sa vykresľuje úsečka po X-ovej alebo Y-ovej osi. Jednotlivé body úsečky sa vypočítajú pomocou prírastku smernice na danú os. Ak je smernica menšia ako 1, vzorkujeme úsečku podľa osi X, ak je sklon väčší ako 1, vzorkujeme podľa osi Y.

Pre vykreslenie kružnice nám postačí uhol 0-45 stupňov, pretože ak bod $A[x, y]$ leží na nej, tak sa na kružnici nachádzajú aj všetky kladné a záporné kombinácie tohto bodu.

3.2 Orezávanie

Pri zobrazovaní objektov na monitore môže nastať situácia, pri ktorej sa celý objekt nenachádza na obrazovke. Preto je dôležité danú úsečku, kruh, či krivku orezať.

Najčastejšie sa používa Cohen-Sutherlandov orezávací algoritmus. Princíp algoritmu spočíva v označení oblastí, v ktorých sa úsečka nachádza. Všetky oblasti sú definované štyrmi číslami, ktoré označujú plochy hraníc obrazu, plochu nad obrazom, pod obrazom, vľavo alebo vpravo od obrazu. Ostatné plochy sú výsledkom kombinácie vyššie uvedených plôch pomocou operátora XOR. Ak sa úsečka nachádza na zobrazovacej ploche, vykreslí sa celá. Ak sa nachádza na viacerých plochách, nastáva rekurzívne delenie veľkosti úsečky, až kým sa daný bod mimo plochy bude nachádzať na hranici zobrazenia.



Obrázok 9: Rozdelenie obrazu na plochy

Ďalší algoritmus na orezanie úsečky sa nazýva Liang Barsky.

3.3 Ray-casting

V tejto metóde sa vysiela svetelný lúč zo zdroja pixel po pixeli podľa smeru (uhla) svetelného zdroja. U každého lúča sa hľadá najbližšie teleso, do ktorého lúč narazil a vyhodnotí sa farba podľa optického modelu (napr. Phongov model).

3.4 Raytracing

Raytracing, po slovensky sledovanie lúča je jedna z metód prevodu 3D reprezentácie scény do 2D zobrazenia. Umožňuje vytvárať značne realistické scény, za cenu vysokého nároku na výkon. Je

vhodnejší pre aplikácie, v ktorých môže byť daná scéna vygenerovaná pomalšie (obrázky, filmové efekty), než pre real-time aplikácie, kde je kladený dôraz na rýchlosť (počítačové hry). Raytracing nebol veľmi rozšírený v začiatkoch vývoja počítačov. Vtedy bola táto metóda príliš náročná na výpočet a počítače nemali dostatočnú rýchlosť. Vytvorené scény neboli kvalitnejšie, než obrazy vytvorené vtedy menej náročnými technikami. Príchodom novej generácie počítačov sa táto metóda začala často využívať a v dnešnej dobe sa používa najmä vo filmovom priemysle, na generovanie špeciálnych efektov.

Raytracing spočíva v tom, že sledovanie lúča neskončí pri nájdení najbližšieho objektu a vyhodnotení jeho farby ako pri ray-castingu, ale svetelný lúč pokračuje ďalej v scéne smerom, ktorý je odvodený podľa odrazivosti a priehľadnosti daného objektu, do ktorého lúč narazil. Lúče, ktoré boli vyslané do scény od pozorovateľa sa nazývajú primárne lúče. Lúč, ktorý je vytvorený po dopade primárneho lúča na objekt a od daného objektu sa odrazí, poprípade lomí, nazývame sekundárny lúč.

Výpočet sekundárneho lúča závisí na materiáli, z ktorého je objekt vytvorený. Ak má predmet odraziť lúč, smer sekundárneho lúča bude vypočítaný podľa známeho fyzikálneho zákona odrazu svetla. Ak má predmet charakteristiku pre lom prichádzajúceho lúča, použije sa Snellova rovnica lomu svetla.

Pri tejto metóde sa používa tretí typ lúča, ktorý voláme tieňový. Tento lúč smeruje z bodu, kam dopadol primárny alebo sekundárny lúč rovno do zdroja svetla a sleduje, či sa lúč nepretína s ďalšími objektami. Ak áno, teleso vysielané tento lúč bude zatienené. Z každého bodu dopadu sa vyšle minimálne toľko lúčov, koľko je zdrojov svetla.

Raytracing algoritmus sa dá vysvetliť nasledujúcim pseudokódom[7]:

Pre každý pixel na obraze:

Vytvor lúč z miesta pozorovateľa cez rovinu pozorovania

Nastav Vzďialenosť_Lúča na nekonečno a Najbližší_Objekt na hodnotu NULL

Pre každý objekt na scéne:

Vypočítaj vzdialenosť priesečníka lúča s objektom v smere lúča

Ak vzdialenosť priesečníka je menšia ako Vzďialenosť_Lúča:

Nastav Vzďialenosť_Lúča na túto vzdialenosť

Nastav Najbližší_Objekt na tento objekt

Pokračuj na ďalší objekt

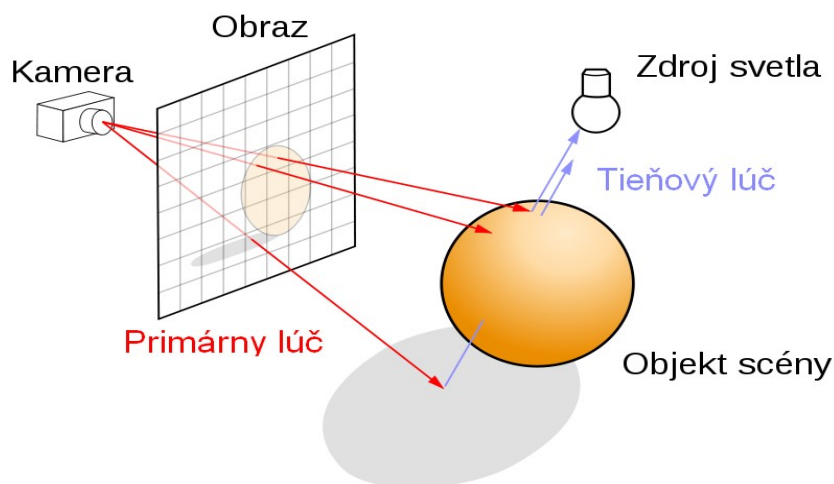
Ak nie je Najbližší_Objekt NULL:

Vypočítaj Farbu tohto pixelu vyslaním tieňového lúča ku každému zdroju svetla

Vytvor sekundárny lúč, ktorý bude na scéne odrazom alebo lomom pokračovať podľa atribútov objektu do ktorého predošlý lúč narazil

Inak :

vyplň pixel farbou pozadia



Obrázok 10: Princíp sledovania lúčov

Ďalšie metódy sú napríklad scanline algoritmus, ktorý je veľmi rýchly, ale ťažko sa s ním pracuje pri CSG (constructive solid geometry), ďalej finite-element, kde sa zo zdrojov svetla „strielajú“ fotóny.

3.5 Vzájomná poloha objektov a lúčov

Aby sa mohla farba daných objektov na scéne vykresliť, je potrebné nejakým spôsobom zistiť prienik svetelného lúča a povrchu objektu. Pre výpočet týchto prienikov je analytický výpočet jeden z najjednoduchších metód.

3.5.1 Prienik dvoch priamok v 2D

Výpočet prieniku dvoch priamok analyticky je veľmi ľahká a rýchla metóda.

Majme parametrické rovnice priamok

$$p: [A + V * t]$$

$$q: [C + U * r]$$

kde A a C sú body priamok, U a V sú ich smerové vektory vypočítané rozdielom bodov $V = [B - A]$, $U = [D - C]$ a r, t sú parametre. Na výpočet prieniku dvoch priamok upravíme predchádzajúce parametrické rovnice na:

$$A + V * t = C + U * r$$

Pomocou jednoduchých úprav

$$(A_x + V_x * t) = (C + U_x * r) \rightarrow *(-V_y)$$

$$(A_y + V_y * t) = (C + U_y * r) \rightarrow *(V_x)$$

$$A_x * (-V_y) + A_y * (V_x) = (C_x + U_x * r) * (-V_y) + (C_y + U_y * r) * (V_x)$$

$$A_x * (-V_y) + A_y * (V_x) + (C_x * (-V_y)) - (C_y * V_x) = +(U_x * -V_y + U_y * V_x) * r$$

vypočítame parameter r

$$r = \frac{(A_x - C_x) * V_y + (C_y - A_y) * V_x}{(U_y * V_x) - (U_x * V_y)} \quad (1)$$

pre parameter t získame podobný vzorec pomocou vyššie uvedených úprav

$$t = \frac{(A_x - C_x) * U_y + (C_y - A_y) * U_x}{(U_x * V_y) - (U_y * V_x)} \quad (2)$$

Tieto parametre nám pomôžu vypočítať súradnice bodu, ktorý pretína dané dve priamky a to pomocou ich dosadenia do parametrických rovníc prvej alebo druhej priamky, podľa toho, či chceme vypočítať bod podľa prvého alebo druhého parametra.

3.5.2 Prienik priamky a kružnice v 2D

Ďalší základný objekt, ktorého prienik je potreba vypočítať je kružnica.

Majme obecnú rovnicu kružnice

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

a parametrické rovnice priamky

$$x = A_x + V_x * t$$

$$y = A_y + V_y * t$$

dosadíme do rovnice kružnice

$$(A_x + (V_x * t) - x_0)^2 + (A_y + (V_y * t) - y_0)^2 = r^2$$

Ak danú rovnicu roznásobíme, získame kvadratickú rovnicu, kde

$$A = V_x^2 + V_y^2$$

$$B=2*(V_x*(A_x-x_0)+V_y*(A_y-y_0))$$

$$C=(A_x-x_0)^2+(A_y-y_0)^2-r^2$$

Riešenie tejto kvadratickej rovnice spočíva v diskriminante

$$D=B^2-4*A*C$$

Ak je diskriminant záporný, rovnica nemá riešenie, čiže priamka kružnicu nepretína. Keď sa diskriminant rovná nule, rovnica má jedno riešenie, priamka pretína kružnicu v jednom bode, ktorý je vyjadrený rovnicou

$$t=\frac{-b}{2*a} \quad (3)$$

Ak diskriminant je kladné číslo rôzne od nuly, priamka a kružnica sa pretína v dvoch bodoch

$$t_{1,2}=\frac{-b\pm\sqrt{D}}{2*a} \quad (4)$$

parameter t potom dosadíme do rovnice priamky a bod prieniku sa vypočíta ako súčet začiatočného bodu lúča (súradnice zdroja) a smerového vektora lúča V vynásobený hodnotou parametra t.

4 Win32 API

Windows API, alebo Win32 API bolo vytvorené spoločnosťou Microsoft a slúži ako jadro aplikačného programového rozhrania (API – application programming interfaces) pre operačný systém Windows. Rozhranie vystavuje programátorovi veľkú časť základnej štruktúry systému Windows. Toto umožňuje vývojárom používať veľké množstvo preddefinovaných funkcií a prejavuje značnú flexibilitu ich aplikáciám. Začiatkom vývoja vyvolalo prostredie Win32 API značný škandál, keďže obyčajný „Hello world“ program obsahoval vyše 150 riadkov a spúšťací skript, v ktorom bolo ďalších 20. Postupom času boli vykonané značné zmeny na systéme Windows, a keďže Win32 API reprezentuje tento operačný systém, rozšírilo sa aj toto rozhranie.

Služby rozhrania Win32 API:

1. Základné služby umožňujú prístup k elementárnym funkciám systému Windows. Patria sem súborové systémy, zariadenia, procesy, vlákna a obsluha chýb.
2. Pokročilé služby dokážu obsluhovať jadro systému. Sú to napr. registre windowsu, vypnutie alebo reštartovanie systému, vytvorenie služby a spravovanie užívateľských účtov.
3. GDI (Graphics Device Interface) je štandard pre reprezentáciu grafických objektov a poskytuje výstup grafického obsahu na monitory, tlačiarne a ďalšie výstupné

zariadenia. Slúži na kreslenie základných grafických objektov, ako sú body, priamky, kružnice, krivky a zobrazenie textov pomocou fontov[10].

4. Užívateľské rozhranie, poskytuje služby pre ovládanie základných prvkov, tlačidiel klávesnice, alebo myši.
5. Knižnica dialógových okien poskytuje základné služby ako uloženie súboru, načítanie súboru, funkcie pre nastavenie farby a písma.
6. Knižnica ovládacích prvkov obsahuje statusbary (stavové lišty), progressbary (postupové lišty), toolbary (nástrojové lišty) a pár pokročilých ovládacích prvkov.
7. Windows Shell, ktorý umožňuje Win32 API prístup do príkazového riadku, takisto zmeniť ho alebo vylepšiť.
8. Sieťové služby umožňujú prístup aplikácie k sieťovému vybaveniu systému. Podkomponenty týchto služieb sú NetBIOS (Network basic input/output system), Winsock (Windows sockets API), NetDDE (Dynamic data exchange) a mnoho ďalších.

5 Analýza a návrh riešenia

Cieľom práce bolo vytvorenie programu, ktorý je založený na sledovaní lúča. Prvým krokom problematiky je vytvorenie scény. Na scénu sa nanesú optické prvky a vytvorí sa simulácia, ktorá zobrazuje sledovanie svetelného lúča, meniaceho svoj smer na základe nárazu na povrch týchto optických prvkov, ako sú napr. zrkadlá a šošovky. Pri tvorbe scény je zrejmé, že tam bude kontakt s koncovým užívateľom. Preto by program mal byť taký, aby užívateľ čo najjednoduchšie pochopil ako sa s oknomarába, vedel skonštruovať jednotlivé objekty a zostaviť scénu.

Základný jazyk v rozhraní volím anglický jazyk, aby bol program použiteľný aj na medzinárodnej úrovni.

Veľkosť scény obmedzím na 1200x600 pixelov. Vstupná scéna bude načítaná do operačnej pamäti podľa dvoch možností

1. zostavená samotným užívateľom
2. načítaná zo súboru, ktorý má presné formátovanie

Ako výstup programu bude zobrazená scéna sledovania lúča a uloženie dát o súradniciach a uhloch primárnych a sekundárnych lúčov. Programovací jazyk zvolím C/C++ s grafickou knižnicou Win32 API. Pred začatím programovania som sa oboznámil s teóriou o optike, o svetle, svetelnom lúči a základných optických prvkoch. Preštudoval som si látku o optike z učebníc fyziky a vlastných poznámok. Ďalej som sa oboznámil s knižnicou Win32 API, v ktorej si vytvorím grafickú časť programu. Aby bolo splnené zadanie, práca má obsahovať druhý program, ktorý ako vstupné parametre bude obsahovať počet odrazov lúčov a súbor na načítanie optických prvkov. Na výstupe tohto programu bude výstup všetkých primárnych a sekundárnych lúčov podľa počtu interakcií lúčov s optickými objektami. Kvôli otočenej Y-ovej osi v súradnicovej sústave vo Windows API bude otočená Y-ová os aj v pomocnom programe. Tento pomocný program slúži na optimalizáciu smeru lúča. Samotná grafická reprezentácia a jadro práce s výpočtami vyriešenými v pomocnom programe bude načrtnutá približne takto.

Najskôr sa vytvorí jednoduché okno, veľkosti 1200x600px, s jednoduchým panelom, v ktorom bude možné otvoriť súbor, uložiť scénu do súboru, pridať jednotlivé optické prvky typu zdroj, rovinné zrkadlo, guľové zrkadlo, spojnú a rozptylnú šošovku. Do sekcie starajúcej sa o pridávanie objektov dám možnosť vytvorenia prvku s presnými údajmi. To umožním vytvorením ďalšieho okna, v ktorom bude zobrazený panel pre výber optického predmetu a textové panely, ktoré budú slúžiť na doplnenie údajov o pozícii, výške a uhle otočenia predmetu. Naďalej tu bude možnosť užívateľovi kedykoľvek zrušiť voľbu vytvorenia predmetu pridaním tlačidiel na potvrdenie, alebo zrušenie pridania. Pre optické prvky na scéne bude realizovateľná možnosť odstránenia, poprípade navrátiť vymazaný objekt na scénu. Ako vo väčšine programov býva vyčistenie obrazovky, zvolil som, že užívateľ nebude musieť všetky objekty jednotlivito mazať zo scény, ale bude mať možnosť odstránenia všetkých prvkov naraz jedným tlačítkom.

Krátka pomôcka pre užívateľa bude v pomocnom paneli. Bude tam krátky anglický popis ako sa s projektom má pracovať, ako vykresliť objekty na scénu a pod.

Pri rozbere problematiky, ako pridávať jednotlivé optické prvky na scénu myšou, bude sa zobrazovať najskôr poloha myši v X-ovej a Y-ovej súradnici, potom veľkosti prvku v pixeloch a nakoniec uhol podľa súradníc objektu a polohy myši na scéne.

Objekty sa v programe budú ukladať ako štruktúra údajov o ich polohe, veľkosti a uhla podľa natočenia. Počet objektov môže byť nekonečný, takže miesto pevného maximálneho počtu objektov bude vytvorený systém, ktorý bude môcť obsahovať teoreticky nekonečné množstvo prvkov. Pre vykreslenie scény sa jednoducho prejde celým zoznamom a podľa údajov sa prvky vykreslia na scénu. Podobný spôsob sa definuje pre primárne a sekundárne lúče. V každom svetelnom zdroji bude údaj primárneho lúča, ktorý sa pretne s povrchom optického prvku a vznikne určitý počet sekundárnych lúčov.

6 Implementácia

Cieľom kapitoly je oboznámiť čitateľa s programom a postupmi, ktoré v ňom boli využité. Implementačný jazyk práce je C/C++ s podporou grafickej knižnice Win32 API. Výhodou týchto jazykov je, že sú ľahko prenositeľné a ľahká zrozumiteľná syntax.

Na základe zadania, projekt nemusí byť multiplatformový, takže bol vytvorený a odskúšaný na operačnom systéme Windows XP Professional. Projekt bol vytvorený v programovom prostredí Microsoft Visual Studio 2008 Professional, s podporou jazyka C/C++. Toto prostredie má v sebe zabudovaný prekladač a príslušného debuggeru.

Podľa zadania bolo jednou úlohou vytvoriť program o sledovaní lúčov, ktoré začínajú v zdroji svetla a majú daný smer pomocou uhla. Ako dátové typy som využil štruktúry pre optické prvky a štruktúry pre lúče. Štruktúry optických prvkov TObject obsahujú údaje o ich identifikačnom čísle, charakteristike, polohe, veľkosti, ohniskovej vzdialenosti a uhle otočenia. Štruktúry lúčov TLObject obsahujú správu o počte odrazov alebo lomov daného lúča, údaj o začiatočnej polohe lúča, informáciu kam lúč dopadol, uhol medzi týmito bodmi, uhol akým bude smerovať po náraze a údaj o prvku, do ktorého narazil. Tieto údaje sa nachádzajú v jednosmernom lineárnom zozname, ktorý sa dynamicky alokuje pri každom novom prvku, alebo lúči. Týmto spôsobom tam užívateľ môže pridať ľubovoľný počet optických prvkov bez zásahu do zdrojového kódu.

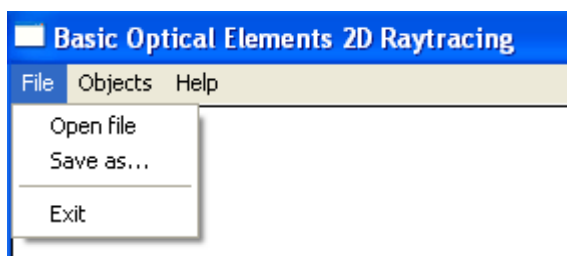
TObject			
Identifikačné číslo objektu			
Definícia objektu (zdroj, zrkadlo, ...)			
Pozícia	<table><tr><td>x</td></tr><tr><td>y</td></tr></table>	x	y
x			
y			
Veľkosť objektu			
Ohnisková vzdialenosť objektu			
Uhol objektu			
Zoznam primárnych a sekundárnych lúčov, ak je objekt zdroj			
Farba lúča, ktorú zdroj vyžaruje			

TLObject			
Číslo lúča			
Identifikačné číslo objektu, do ktorého lúč narazil			
Definícia objektu, do ktorého lúč narazil			
Začiatočná pozícia	<table><tr><td>x</td></tr><tr><td>y</td></tr></table>	x	y
x			
y			
Koncová pozícia	<table><tr><td>x</td></tr><tr><td>y</td></tr></table>	x	y
x			
y			
Premenná pre kontrolu skončenia výpočtu			
Veľkosť parametra prieniku			
Uhol predošlého lúča			
Nový uhol lúča			

Obrázok 11: Zobrazenie datových štruktúr optických objektov a lúčov

Bakalárska práca je zložená z dvoch programov. Prvý program som vytvoril ako konzolovú aplikáciu, ktorá slúži na výpočet smeru odrazených a lomených lúčov. Vstupné argumenty tohto softvéru sú číslo, ktoré udáva počet interakcií lúčov s optickými prvkami a názov súboru, z ktorého sa načíta daná scéna so svetelnými zdrojmi a objektami. Výstup tejto aplikácie bude počet lúčov, ktoré sa odrazili alebo lomili na objektoch. Program slúži ako pomocný nástroj na optimalizovanie optických prvkov na scéne pre zmenu smeru lúča. Algoritmy na výpočet odrazov a lomov sú popísané ďalej.

Scénu som vytvoril ako biele okno s rozmermi 1200x600px. Pre komunikáciu s užívateľom sa v okne nachádzajú tri základné panely. V prvom paneli si môže užívateľ vybrať jednu z možností načítania scény zo súboru, uloženia scény a ukončenia programu.



Obrázok 12: Zobrazenie menubaru File

Súbory uložené alebo načítané musia mať presné formátovanie. Prvý riadok súboru obsahuje počet prvkov, ktoré sa načítajú, na ďalších riadkoch sa nachádzajú jednotlivé objekty opísané pôvodom objektu, súradnicami, výškou, ohniskovou vzdialenosťou a uhlom otočenia.

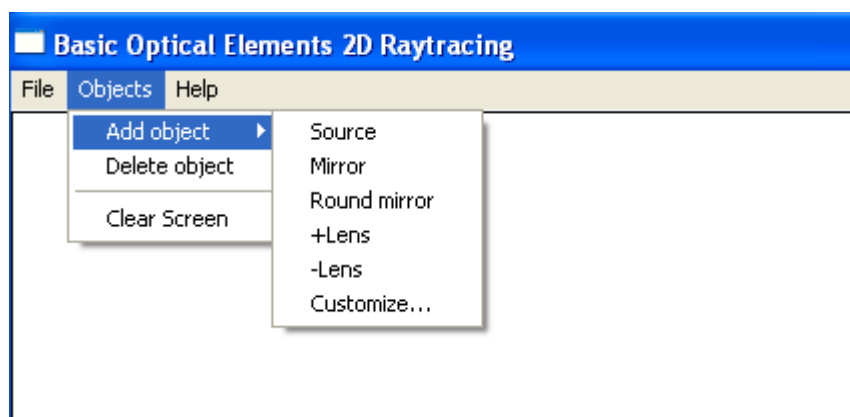
Druhý panel predstavuje prácu s objektami, pridanie optického prvku, vymazanie, poprípade vyčistenie celého obrazu. Pridávanie obrazu sa môžeme rozdeliť na dve možnosti. Prvá možnosť pridania je funkcia Customize, ktorá vytvorí dialógové okno, kde si užívateľ vyberie objekt a potom nastaví údaje o polohe prvku, veľkosti, v prípade šošoviek ohniskovej vzdialenosti a nakoniec uhla otočenia na scéne. Druhá možnosť vykreslenia objektu na scénu je realizovaná myšou. Táto možnosť spočíva v dvoch, troch, až štyroch fázach.

Vytvorenie zdroja svetla má dve fázy. V prvej sa nastaví polohou myši súradnice zdroja, stlačením ľavého tlačítka sa súradnice uložia a prechádza sa do druhej fázy, kde sa nastavuje uhol otočenia zdroja a smer vyžarovaného lúča.

Pri zrkadlách sú tri fázy. Najskôr sa nastaví pozícia podľa myši. Prvým kliknutím na myš sa uložia súradnice objektu a nastáva prechod do druhej časti. V tejto časti sa nastaví výška, poprípade dĺžka objektu. V tretej fáze sa nastaví uhol a nakoniec sa objekt uloží do zoznamu.

Šošovky obsahujú štyri časti. Prvá časť a druhá časť spočíva, ako pri zrkadle, nastavením polohových súradníc stredu šošovky a jej dĺžky. V tretej fáze sa nastaví ohnisková vzdialenosť šošovky, nakoniec uhol otočenia na scéne a prvok sa uloží do zoznamu.

Pre zjednodušenie pridávania objektov na scénu som vytvoril funkciu, ktorá slúži na výpis údajov pod kurzor myši. V základnom stave je pod kurzorom súradnica, kde sa nachádza. V ďalších fázach pridávania objektov sa na pod ním nachádzajú príslušné údaje o veľkosti objektu, ohniskovej vzdialenosti, poprípade uhla otočenia.



Obrázok 13: Zobrazenie menubaru Objects

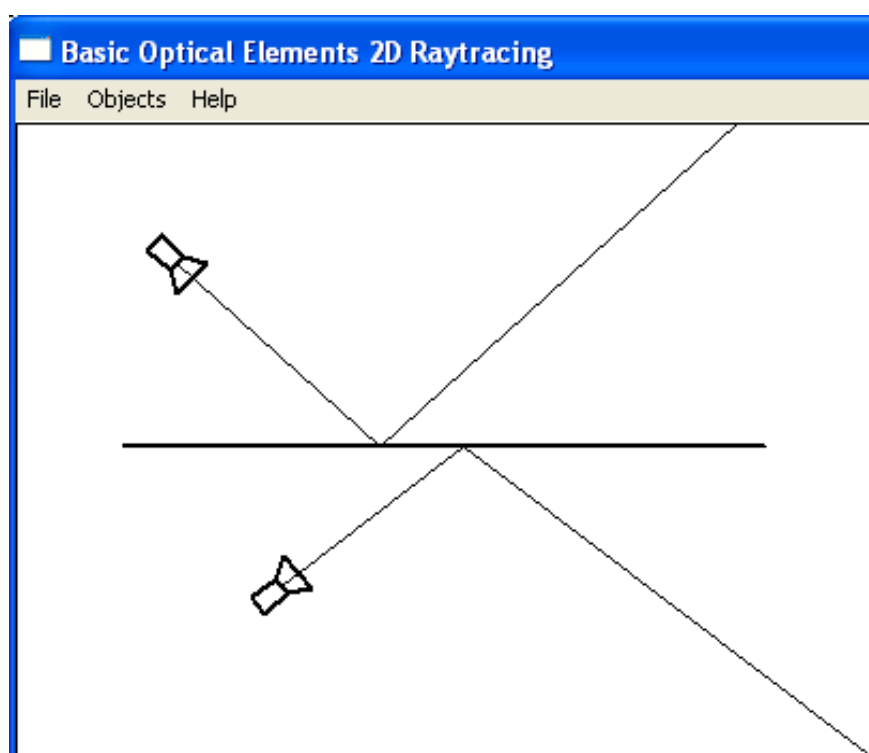
Na vymazanie jednotlivých objektov slúži menubar Delete object. Táto funkcia zobrazuje dialógové okno, ktoré na ľavej strane predstavuje objekty na scéne a na pravej strane vymazané objekty, aby pri chybnom vymazaní prvku bola užívateľovi navrhnutá možnosť vrátiť prvok späť na scénu. Pri uzavretí tohto okna, všetky prvky nachádzajúce sa na pravej strane zoznamu budú

vymazané a nebudú môcť byť už spätne navrátené. Pri konštrukcii scény môže užívateľ kedykoľvek pridať alebo odstrániť objekt.

Akonáhle sa na scéne zobrazí prvý zdroj lúča, vysiela sa z neho lúč do scény a začnú sa počítať odrazy a lomy tohoto lúča, na základe objektov na ktorých povrch narazil. Tento výpočet sa vykoná známym raytracing algoritmom (kap. 3.4).

Tematikou tejto práce je zaoberať sa smerom šírenia lúča, takže vykreslenie povrchu objektov podľa farby lúča nebolo podstatné. Objekty majú čiernu farbu a sú vykreslené na scéne čo najzrozumiteľnejšie a najjednoduchšie.

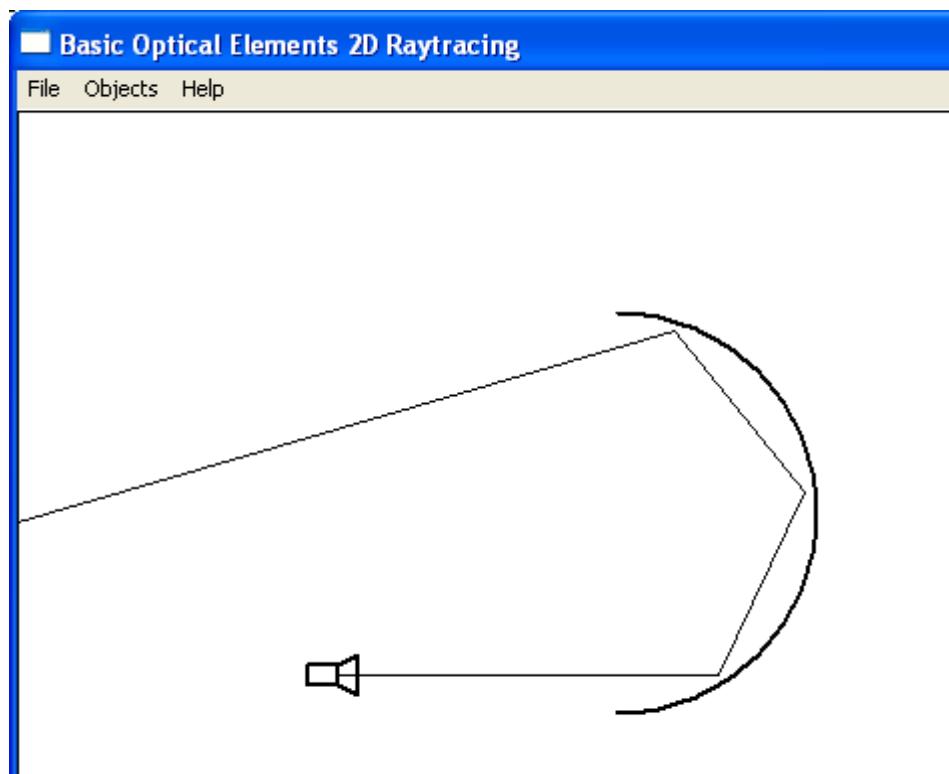
Ak lúč narazí na objekt rovinného zrkadla, je vypočítaný nový lúč podľa vzorca vyvedeného zo zákona odrazu. Prienik lúča a zrkadla je daný v bode, vypočítanom podľa prieniku dvoch priamok. Vzorec, ktorý som využil pre výpočet riešenia sa nachádza v kapitole 3.5.2, označený číslom 1.



Obrázok 14: Zobrazenie riešenia odrazu lúča cez rovinné zrkadlo

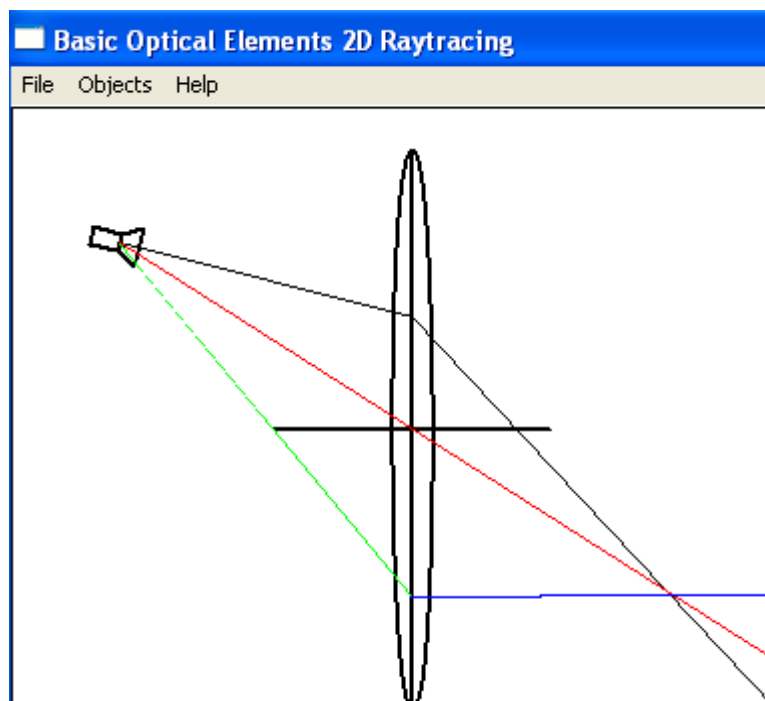
Pre výpočet smeru odrazeného lúča z guľového zrkadla platí ten istý algoritmus ako pre rovinné. Výpočet prieniku plochy guľového zrkadla a lúča je vyvedený z kapitoly 3.5.3, vzorec číslo 3 a 4. Ak sa lúč s daným guľovým zrkadlom pretne v určitom bode, vypočíta sa normálový uhol zrkadla určený bodom prieniku lúča s daným objektom a stredom tohto prvku. Následne sa aplikuje výpočet smeru odrazeného lúča pomocou uhla dopadajúceho lúča a vyriešeného normálového uhla. Pri riešení tejto situácie nastal problém, že daný lúč pretiekol cez povrch zrkadla, čo sa prejavilo tak, ako by sa lomil a nie odrazil. Túto situáciu som vyriešil nasledovne. Vypočítal som všetky potrebné údaje, aby som zistil smer daného odrazeného lúča. Akonáhle som získal výsledky, lúč som nastavil

tak, aby bod prieniku nebol presne v guľovom zrkadle, ale niekoľko bodov pred ním. Týmto spôsobom som vyriešil spomenuté pretekánie lúča zo zrkadla nesprávnym smerom.



Obrázok 15: Zobrazenie odrazu lúča cez guľové zrkadlo

Situáciu lomu lúča pri prechode šošovkami som vyriešil pomocou zobrazovacích rovníc predmetov a obrazov popísaných v kapitole 2.4.5. Ako znázorňuje obrázok 13, lúč pretína šošovku v bode, ktorý je zobrazený čiernou priamkou vedúcou zo zdroja svetla do šošovky. Mojm prvým krokom pre výpočet smeru lomeného lúča je vytvorenie priamky zo zdroja, ktorá prechádza stredom šošovky (priamka červenej farby). Ďalej vytvorím úsečku, ktorá má začiatkové súradnice v zdroji a bude prechádzať ohniskom šošovky (na obrázku má zelenú farbu). Následne zistím bod, ktorý vznikol prienikom zelenej úsečky so stredovou čiarou optického prvku. Z tohto bodu vytvorím priamku, ktorá má smernicu vypočítanú podľa kolmice na uhol otočenia šošovky na scéne (modrá priamka). Nakoniec vyriešim prienik modrej priamky, s úsečkou, ktorá prechádza stredom šošovky (červená farba). Daný prienikový bod volím ako smerový vektor novovzniknutého lomeného lúča. Ak bola vzdialenosť zdroja svetla od šošovky menšia ako ohnisková vzdialenosť, vypočítal sa nový uhol lomu tak, že som najskôr vytvoril priamku kolmú na šošovku a zistil som bod, kde sa dané priamky pretínajú. V tomto bode som skonštruoval úsečku, ktorá prechádza druhým ohniskom šošovky. Bod pretnutia tejto úsečky a priamky, ktorá prechádza zdrojom svetla a stredom šošovky použijem ako smerový vektor lomeného lúča, z ktorého vypočítam uhol.



Obrázok 16: Zobrazenie lomu lúča pri prechode spojnou šošovkou

Pri týchto výpočtoch nastal tiež problém, že daný odrazený alebo lomený lúč zostával v optických prvkoch. S touto udalosťou som si poradil nasledovne: do každého nového lúča som pridal údaj o identifikačnom čísle optického prvku, do ktorého narazil. Týmto spôsobom sa výpočet lúča nemohol opakovať pre ten istý predmet dva krát za sebou, ale až kým lúč nenarazil do iného objektu.

Pri testovaní programu som prišiel na množstvo chýb, ktoré vznikli zadaním hraničných hodnôt, ako napr. zdroj svetla sa nachádzal presne v ohniskovej vzdialenosti od šošovky, alebo bol rovnobežný so stredom šošovky. V týchto prípadoch sa vzniknuté priamky pretli v nekonečne a nebolo možné vypočítať lomený uhol. Túto situáciu som vyriešil pomocným zdrojom svetla, ktorý má ten istý smer ako zdroj skutočný, ale jeho pozícia sa posunula o niekoľko bodov ďalej, aby bolo možné výpočet realizovať.

7 Záver

Cieľom tejto bakalárskej práce bolo naštudovať teóriu o optike, svetle a jednoduchých optických prvkoch, navrhnúť a implementovať metódu sledovania lúča pri prechode optickým prostredím s ohľadom na možnosť rozšírenia programu o nástroj, ktorý bude optimalizovať optickú sústavu. Tento cieľ bol splnený.

Túto tému som si vybral, pretože ma počítačová grafika zaujala a chcel som sa v nej zdokonaľiť. Po preštudovaní literatúry o svetle a optických prvkoch som vytvoril tento jednoduchý program, ktorý zobrazuje svetelný lúč a jeho interakciu s jednotlivými optickými objektami. Počas štúdia tejto problematiky som hľadal rôzne metódy a praktiky, ktoré by znázorňovali podobnú situáciu, ale bohužiaľ som nenašiel žiadne riešenia podobnej tematiky. Program má uplatnenie najmä v oblasti fyziky, môže slúžiť ako didaktická pomôcka. Keďže jazyk rozhrania je anglický, je projekt použiteľný na celom svete.

7.1 Možnosti rozšírenia

Keďže v našom svete existuje nekonečne veľa predmetov, lúče dopadajú na povrch každého z nich, znamená to, že je nekonečne veľa možností ako prácu rozšíriť o tieto objekty. Nový optický prvok môže byť definovaný ako prostredie ohraničené v mnohouholníku, napr. voda, sklo a iné priesvitné látky. Ako ďalší prvok by sa mohol pridať objekt, ktorý absorbuje určitú, poprípadne celkovú energiu lúča. Ak by do tohto objektu lúč narazil, jeho výpočet by sa skončil, alebo by sa obmedzil (pohltil) jej počet interakcií s ďalšími prvkami.

Ďalšie možné vylepšenie na projekte je napríklad optimalizácia výpočtu, ako napr. delenie na podpriestory, alebo paralelizácia výpočtov [11].

Jedno z ďalších vylepšení je vytvorenie polopriepustnosti, kde by sa jedna časť lúča odrazila, druhá by sa lomila do objektu a tieto novovzniknuté lúče by pokračovali s polovičnou energiou (polovičný počet ďalších možných odrazov a lomov).

Literatúra

- [1] Samuel Krupka, Odraz, lom lúča na optických prvkoch, Univerzita Pavla Jozefa Šafárika v Košiciach
<http://s.ics.upjs.sk/~random/skola/df/u5.pdf>
- [2] Skriptá o optike, časť 2_4, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave
http://fyzikus.fmph.uniba.sk/typo/uploads/media/F4G_2_4.pdf
- [3] Skriptá o optike, časť 2_5, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave
http://fyzikus.fmph.uniba.sk/typo/uploads/media/F4G_2_5.pdf
- [4] Skriptá o optike, časť 2_6, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave
http://fyzikus.fmph.uniba.sk/typo/uploads/media/F4G_2_6.pdf
- [5] Jana Pílniková, Počítanie prienikov, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave
<http://www.sccg.sk/~pilnikova/pg/prieniky.pdf>
- [6] Vzájomná poloha geometrických útvarov, Katedra matematiky a deskriptívnej geometrie, SvF STU.
<http://www.math.sk/skripta/node35.html>
- [7] fuzzyphoton.tripod.com: Raytracing algorithm
<http://fuzzyphoton.tripod.com/rtalgo.htm>
- [8] Wikipédia: Windows API
http://en.wikipedia.org/wiki/Windows_API
- [9] Zetcode.com: The Winapi (C Win32 API, No MFC) tutorial
<http://zetcode.com/tutorials/winapi/>
- [10] Wikipédia: Fonts
<http://sk.wikipedia.org/wiki/Font>
- [11] Don Fussell, Anti-aliased and accelerated ray tracing CS384G - Computer Graphics, University of Texas at Austin, Spring 2010:
http://userweb.cs.utexas.edu/~fussell/courses/cs384g/lectures/lecture10-Aa_and_accel_raytracing.pdf

Zoznam príloh

Príloha 1. Návod na použitie

Prvým krokom je inštalácia Microsoft Visual Studia. Súbor, ktorý nainštaluje na počítač toto prostredie sa nachádza na CD v priečinku Visual Studio Express. Pre inštaláciu rozhrania je potrebný prístup na Internet. Na kroky inštalácie je použitý tzv. inštaláčny wizard, ktorého kroky treba splniť.

Demonštrácia sledovania lúča je program, vytvorený v prostredí Microsoft Visual Studio. Aplikácia sa dá spustiť otvorením tohto prostredia a načítaním projektu pomocou súboru s názvom Bakalarka.sln v priečinku Bcgraph. Po preložení sa vytvorí spustiteľný súbor Bakalarka.exe, ktorý je výsledným súborom bakalárskej práce. Užívateľské rozhranie je čo najjednoduchšie a je popísané v kapitole 6.

Podobne druhý program sledovania lúča, ktorý slúži ako pomôcka pre rozšírenie o nástroj na optimalizáciu lúča sa preloží otvorením súboru Bc_console.sln, nachádzajúci sa v priečinku Bcconsole. Po preložení tohto súboru vznikne aplikácia Bc_console.exe. Tento program sa spustí pomocou príkazového riadku, podľa nasledovného formátovania:

```
Bc_console.exe <počet_vypísaných_riadkov> <názov_vstupného_súboru>
```

Program načíta scénu do pamäte a vykoná výpočet takého množstva odrazov a lomov, aké je číslo <počtu_vypísaných_riadkov>. Dané riešenia vypíše na štandardný výstup.

Príloha 2. Zdrojové kódy a spustiteľné súbory

Zdrojové súbory sa nachádzajú v CD prílohe v priečinku Sources. Aplikácie sú uložené v priečinku Applications.

Príloha 3. Obsah priloženého CD

priložené CD obsahuje:

1. všetky súbory potrebné na preloženie bakalárskej práce
2. zdrojové kódy práce
3. preložené spustiteľné súbory na zobrazenie demonštrácie
4. text bakalárskej práce vo formátoch *.odt a *.pdf
5. súbory s údajmi objektov, ktoré sú uvedené ako príklady pre načítanie scény. Tieto súbory sa môžu načítať spustenou grafickou reprezentáciou bakalárskej práce.